

**Exhibit C4**

# RINGMASTER 2.0 FUNCTIONAL SPECIFICATION

---

*PROJECT NAME “WHATEVER”*

**Revision 0.12**



---

AUTHORS: RingMaster Engineering

### **Trapeze Networks Proprietary**

The information in this document is confidential and proprietary. This document is the property of Trapeze Networks and shall not be reproduced or copied or used in whole or in part without written permission. This is an unpublished work protected under the Federal copyright laws.

Copyright © [REDACTED] by Trapeze Networks, Inc.  
All Rights Reserved.

<b>1</b>	<b>INTRODUCTION.....</b>	<b>5</b>
1.1	GOALS AND SCOPE.....	5
<b>2</b>	<b>NEW DEVICE SUPPORT (MX-6, MX-400).....</b>	<b>6</b>
2.1	OVERVIEW .....	6
2.2	CONFIGURATION MANAGEMENT.....	6
2.2.1	<i>DTD Changes.....</i>	6
2.2.2	<i>Model Changes.....</i>	6
2.2.3	<i>VersionING.....</i>	6
2.2.4	<i>MX Creation – Local.....</i>	7
2.2.5	<i>MX Creation – Upload.....</i>	7
2.2.6	<i>MX Creation – Other (Open Plan, Import, Paste, etc.).....</i>	7
2.2.7	<i>Chassis Modification.....</i>	8
2.2.8	<i>network synchronization.....</i>	8
2.2.9	<i>Deploy &amp; Distribute Image/Config.....</i>	8
2.2.10	<i>XML Mapping Implementation Notes.....</i>	9
2.2.11	<i>Copy/Paste/Paste Replace.....</i>	9
2.2.12	<i>Configuration File Import.....</i>	9
2.2.13	<i>Configuration File Export.....</i>	10
2.2.14	<i>Verification/Rules Engine.....</i>	11
2.3	IMAGE MANAGEMENT.....	11
2.3.1	<i>Image File .....</i>	11
2.3.2	<i>Image Repository.....</i>	11
2.3.3	<i>Image Configuration .....</i>	12
2.4	PERFORMANCE MANAGEMENT .....	12
2.4.1	<i>Performance Data Aggregation .....</i>	12
2.5	FAULT MANAGEMENT.....	12
2.5.1	<i>Operational Status Monitoring.....</i>	12
2.6	POLICY MANAGEMENT .....	12
2.7	RF PLANNING .....	13
2.8	RF DETECTION.....	13
2.9	REPORTS .....	13
2.9.1	<i>Network Topology Verification .....</i>	13
2.9.2	<i>inventory report.....</i>	13
2.9.3	<i>Work Order .....</i>	13
2.10	ESTIMATES.....	14
<b>3</b>	<b>DISTRIBUTED MP (DMP) SUPPORT .....</b>	<b>15</b>
3.1	OVERVIEW .....	15
3.1.1	<i>Feature Summary.....</i>	15
3.1.2	<i>Distributed Mobility Point (DMP) .....</i>	15
3.2	USE CASES.....	17
3.2.1	<i>Usage of RF Planning Tool.....</i>	17
3.2.2	<i>Deploying DMP configurations.....</i>	17
3.2.3	<i>Managing DesiGn Constraints.....</i>	17
3.2.4	<i>Create a New Distributed MP .....</i>	18
3.2.5	<i>Managing MP Connection InformatioN.....</i>	18
3.2.6	<i>Upload DMP Configuration.....</i>	18
3.2.7	<i>CLI Import/Export .....</i>	18
3.2.8	<i>Copy/Paste/Paste-Replace .....</i>	18
3.2.9	<i>Reboot Dialog.....</i>	18
3.2.10	<i>Inventory Report .....</i>	18
3.3	INFORMATION MODEL .....	19

3.3.1	<i>Distributed MP</i> .....	19
3.3.2	<i>Mobility Point</i> .....	19
3.3.3	<i>Design Constraints</i> .....	20
3.3.4	<i>Coverage Area</i> .....	21
3.4	ENHANCEMENTS TO RF PLANNING .....	21
3.4.1	<i>Design Constraints Management</i> .....	21
3.4.2	<i>MP Computation</i> .....	24
3.4.3	<i>Workorder</i> .....	25
3.5	OTHER ENHANCEMENTS .....	25
3.5.1	<i>Tree View</i> .....	25
3.5.2	<i>Create DMP Wizard</i> .....	27
3.5.3	<i>Edit MP Wizard</i> .....	28
3.5.4	<i>Chassis Wizard</i> .....	30
3.5.5	<i>Reboot Dialog</i> .....	31
3.5.6	<i>Inventory Report</i> .....	31
3.6	FAULT MANAGEMENT.....	32
3.7	IMPACT ON RF DETECTION.....	32
3.8	NETWORK TOPOLOGY VERIFICATION .....	32
3.8.1	<i>Verify Unconfigured MPs</i> .....	32
3.8.2	<i>Verify configured MPs Not Reporting Status</i> .....	33
3.8.3	<i>Verify Mis-configured MPs</i> .....	34
3.8.4	<i>Verify Redundancy Configurations</i> .....	34
3.8.5	<i>Verify Serial Number configuration</i> .....	34
3.9	VERIFICATION RULES.....	34
3.10	CLI MAPPING/DTD CHANGES .....	35
3.10.1	<i>CLI Commands</i> .....	35
3.10.2	<i>DTD Model Changes</i> .....	36
3.10.3	<i>DTD CHANGES for Announce Status</i> .....	37
3.11	STATISTICS .....	37
3.12	IMPACT OF MX VERSIONING .....	37
4	POLICY MANAGEMENT .....	38
4.1	CURRENT DESIGN & ISSUES.....	38
4.1.1	<i>Support for Multiple Device Types &amp; Versions</i> .....	38
4.1.2	<i>Policy Criteria</i> .....	38
4.1.3	<i>Choosing What to Apply</i> .....	38
4.2	PROPOSED CHANGES .....	38
4.3	QUESTIONS & ISSUES .....	39
4.4	CREATE POLICY .....	40
4.5	MODIFY POLICY .....	42
4.6	DELETE POLICIES .....	43
4.7	ASSOCIATE DEVICES & POLICIES .....	43
4.8	APPLY POLICIES .....	43
4.9	CREATE POLICIES FROM DEVICE DATA .....	44
4.10	READING OLD PLANS – BACKWARD COMPATIBILITY .....	44
4.11	IMPROVE PERFORMANCE .....	44
4.12	DELIVERABLES & ESITMATES .....	45
4.13	NOTES ON IMPLEMENTATION .....	45

Revision	Who	Date	Description
0.1	Allan		Original
0.2	Sudhir		Begin DMP support section
0.3	Kishan		Updated impact RF planning section
0.4	Kishan		Updated RF Planning section
0.5	Kishan		Updated RF planning section
0.6	Sudhir		Updates to DMP section
0.7	Jim		MX-6 details
0.8	Sudhir		Added screen shots to depict management of MP connection information
0.9	Jim		Screen shots & updates for MX-6 CM & image sections
0.10	Jim		Updates from review with Allan
0.11	Sudhir		Merged back the working copy
0.12	Kishan		Updates from review with Allan

---

## 1 INTRODUCTION

---

### 1.1 GOALS AND SCOPE

The goal of this document is provide a functional specification of the changes to RingMaster for Version 2.0.

The following main features are targeted for 2.0:

- MX-6/MX-400 Support
- Intermediate L2/L3 MP Support
- Policy Management Changes

---

## 2 NEW DEVICE SUPPORT (MX-6, MX-400)

---

### 2.1 OVERVIEW

The MX-6 is a smaller version of the MX-20, with 8 ports (6 fast Ethernet and 2 gig-ethernet.) The MX-400 is a 4 gig-port chassis. From a software perspective the MX-6 & MX-400 are the same as the MX-20. Please refer to the appropriate PDDs for more product details. These are available at:

<http://intranet.trpz.com/highwire/productmgmt/PDD/v2.0/>

The following sections elaborate on the areas of Ringmaster that need to be changed to handle the new MX types.

### 2.2 CONFIGURATION MANAGEMENT

#### 2.2.1 DTD CHANGES

~~This needs to be co-ordinated with the NOS team.~~

The DTD needs to be modified to have a chassis type attribute as part of the boot status. Ringmaster will use this attribute wherever it needs to check the network type (e.g. Topology reports, deploy, upload, etc.) This is similar to how the version is read & processed today.

#### 2.2.2 MODEL CHANGES

There are no new classes for the new device types. The existing Chassis class and the existing Network Plan -> Device relation will be used to model MX-6 & MX-400 instances. Note that this implies that the chassis names are unique across all types of chassis.

There is currently a "MX Model" RO attribute on the device, that displays the system description value. This will now be used as a RC attribute that allows the user to select an MX model. The system description will be shown in the SNMP properties (as it is also done today.) The MX model will be an NMS only attribute i.e. not part of a deployable config.

New Device Descriptors will need to be created for each type of MX.

#### 2.2.3 VERSIONING

The MX-6 & MX-400 will only allow v2.0 and up (see table).

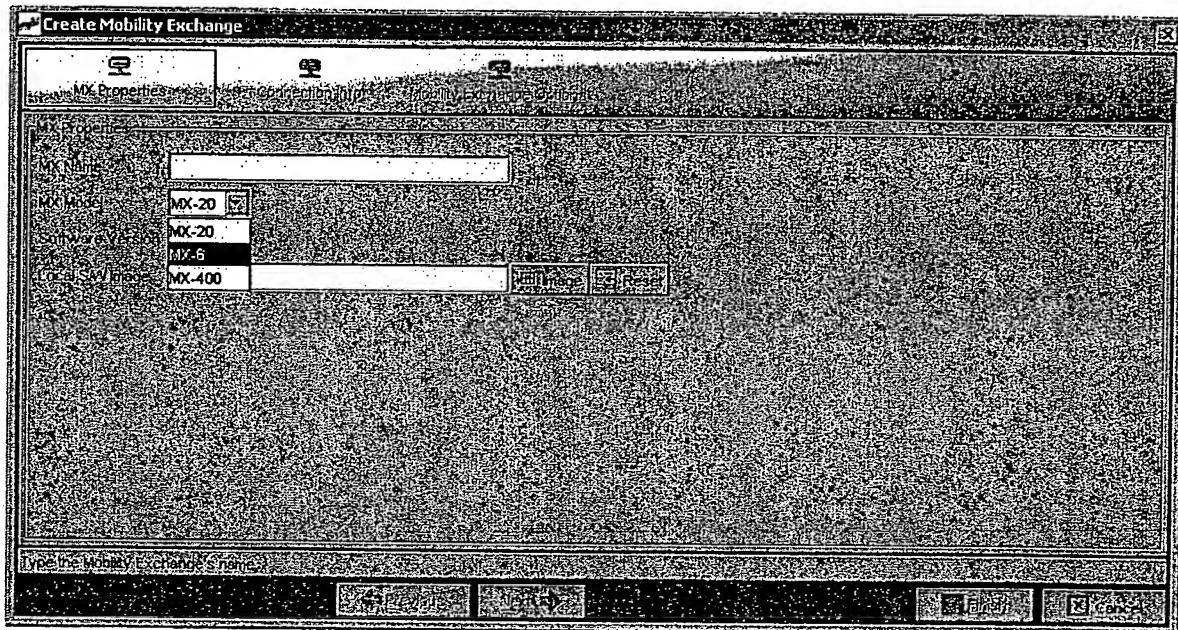
	1.0	1.1	2.0
MX-20	Y	Y	Y
MX-6	N	N	Y
MX-400	N	N	Y

# BEST AVAILABLE COPY

Since the same model classes are used for all types of chassis', the allowed configurable software versions for the MX-6 & MX-20 will be different based on the instance type (controlled via "getValidChoices".)

## 2.2.4 MX CREATION – LOCAL

Users will now need to specify which type of chassis they wish to create. Based on the selected option, the right Device Descriptor will be used to create the default objects for the chassis.



Internally, when the chassis type is selected a default Chassis object is created and set as the context.

*NOTE: If the user comes back to this page and selects a different MX type, the context will be deleted and re-created.*

## 2.2.5 MX CREATION – UPLOAD

During an upload, Ringmaster will determine what type & version of chassis to create based on the system boot status returned from the MX. Once a chassis is created, the XML config is mapped on to it.

## 2.2.6 MX CREATION – OTHER (OPEN PLAN, IMPORT, PASTE, ETC.)

When opening a plan, Ringmaster will use an NMS-only "chassis-type" attribute to determine the type of chassis to create. The same approach will be used for a paste.

For an import or a paste-replace where the device already exists, its type will not be changed but the data will be applied. More details on this in subsequent sections.

# BEST AVAILABLE COPY

## 2.2.7 CHASSIS MODIFICATION

The user will *not* be allowed to change the type of a chassis after it is created (i.e. the create wizard is finished.) Within the create wizard the user can go back to the chassis selection page and start over.

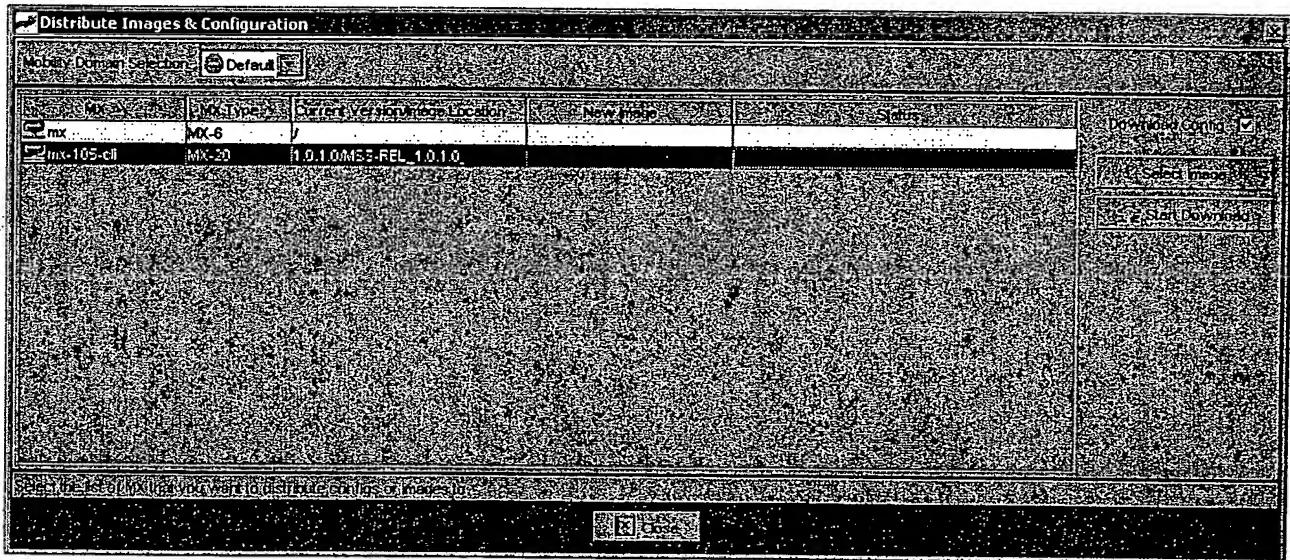
## 2.2.8 NETWORK SYNCHRONIZATION

RingMaster will only allow network changes to be applied to the plan if the type of the MX in both the network and RingMaster configuration are the same. If network changes are detected, but the types are different, the Network Status field in the Local & Network Changes view will indicate the model mismatch and the apply button will be disabled for that MX.

## 2.2.9 DEPLOY & DISTRIBUTE IMAGE/CONFIG

When sending configuration and images to the network, Ringmaster will check and verify the device type. This will be done along with the existing license & version checks, before any configuration is sent. If there is a mismatch, an error will be shown and the operation will fail.

The image distribution and configuration page will need to be modified in way that the image selection button is disabled whenever the user selects multiple MX's of different types. A new MX type column will be added to help the user properly select multiple MX's.



### 2.2.9.1. BUTTON/UI CHANGES

Both the Deploy & the Distribute Images & Configuration have some UI issues.

The Deploy page has a button on the top right. This should be moved to the side or the bottom. The Distribute Images & Config has the "Start Download" button on the side. This should be moved to the bottom panel.

## 2.2.10 XML MAPPING IMPLEMENTATION NOTES

This behaviour is common to all functions like copy & paste/paste-replace, import, upload, etc.

Although a device type cannot be changed via parsing XML, any configuration can be parsed onto any type of device. Hence, the XML mappers will have to be flexible enough to handle parsing of data that may not be completely valid in a best-effort manner.

The main issue is with port references, which can be used in VLANs, ACL Maps, etc. So if a MX-20 VLAN containing port references to port 10 is copied to an MX-6 the expectation is that the VLAN will be properly copied but the VLAN-PORT that is invalid in the target will be ignored. The way this can be handled is if the key reference is not resolved, the mappers do not create the containing object (like VLAN-PORT.)

## 2.2.11 COPY/PASTE/PASTE REPLACE

RingMaster provides useful features to allow the user to copy/paste/paste replace configurations within the supported configuration elements of the network. It is required for the user to be able to copy/paste/paste-replace between heterogeneous MX types. That is, the user should be able to copy and paste a VLAN from an MX-20 to an MX-6. Similarly, a user should be able to paste-replace from an MX-20 to an MX-6 without any significant problems.

The device type attribute will not be applied as the type of an existing device cannot be changed via a copy & paste-replace. For a paste that creates a new device, the type will be the same as the source device.

Just as with copying and pasting across versions, copying and pasting across types will be best effort. This means that only data that is valid in the target will be applied. So, in the VLAN example above if the source VLAN contains port references that are not valid in the target device, these will not be created.

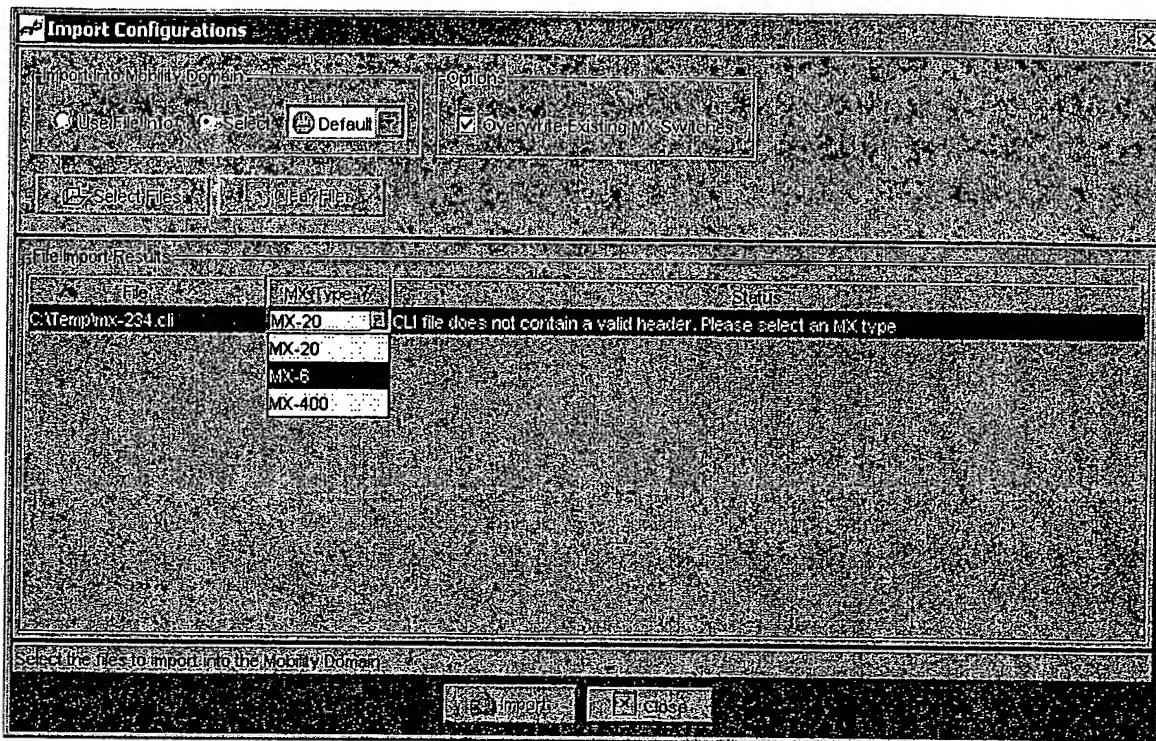
For ports, the configuration will be aligned based on port type and port number. This means that based on the incoming chassis type, the target chassis type, and the incoming port number there will be a best effort attempt to calculate the target port and apply the configuration to it.

On a paste-replace/paste, it is desirable to show the user what configuration was not applied. This could be done as a final progress page of the edit wizard that is displayed after the paste. However, there is no good way to determine what data was not transferred. Needs more thought/research.

## 2.2.12 CONFIGURATION FILE IMPORT

The CLI-based configuration must include a type discriminator to allow RingMaster to build the correct chassis type. This has traditionally been done by a standard comment field at the top of the CLI-based file. We need to define such a standard and incorporate support for this into RingMaster.

When a file is selected, and if it is a CLI file Ringmaster will parse the header and determine the type. If the header is missing, or not properly formatted the user will need to select the device type. For XML files, Ringmaster will use the number & type of ports to determine the chassis type. In either case, the user can always change the device type.



### 2.2.12.1. OVERWRITE EXISTING MX SWITCHES

Currently, the overwrite option really does a merge. This will be modified to make the device config look exactly as in the imported file. *This implies that a partial configuration can no longer be applied to an existing device.*

If the overwrite option is selected, and the MX types do not match the import will fail.

### 2.2.12.2. CLI HEADER FORMAT

~~This needs to be co-ordinated with the NOS team.~~

Here is a proposal for the CLI header (only the "Model" line is new):

```
# Configuration ngen'd at [REDACTED] 15:32:08
# Image 1.1.0.67
# Model MX-400
# Last change occurred at [REDACTED] 18:12:12
```

### 2.2.13. CONFIGURATION FILE EXPORT

RingMaster provides the user with the ability to export CLI-based configuration files. It must be extended to enable creation of configuration files for the MX-6 & MX-400, including the standard type header for the file that specifies the MX type.

### 2.2.13.1 VERIFICATION ON EXPORT

Currently we do not run rules on export of configuration. When the user selects the export option, just as in the deploy wizard, we should run the verification rules and display any errors. The display could be done as a separate "View Errors/Warnings" button on the export dialog. The same preferences that are used today will be used to control whether a configuration with errors can be exported etc.

It would be nice to show the error on a per device basis, and only enable/disable export of that device. This would require some modifications to the verification engine as currently there is no way to request verification on certain sub-trees.

### 2.2.14 VERIFICATION/RULES ENGINE

Other than the physical port count, Ringmaster will not impose configuration limits on VLANs, ACLs, etc.

Ringmaster will have logic to restrict the Distributed AP count for different types of MXs. This is covered in the section for the L2/L3 MP support.

## 2.3 IMAGE MANAGEMENT

### 2.3.1 IMAGE FILE

~~This needs to be co-ordinated with the NOS team.~~

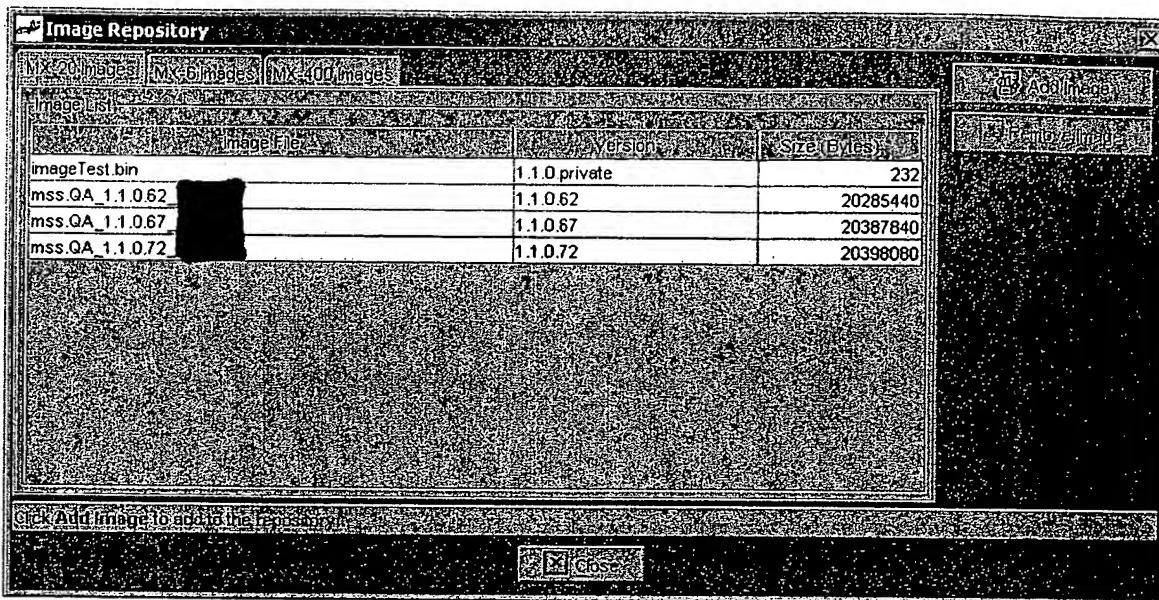
The embedded XML header in the image file will need to be modified to include a device type. The image file parser will read this and hence be able to determine what type of chassis an image is for. Ringmaster will also need to handle the case where the model is missing i.e. for 1.0 and 1.1 images. These will be assumed to be MX-20 images. (The case where a 2.0+ image is missing this information is an error.)

The proposed format is:

```
<image-identifier product="MX" model="MX-20" version="1.1.0.76
label="QA_1.1.0.76_". filename="MX010100.020"> </image-identifier>
```

### 2.3.2 IMAGE REPOSITORY

The image repository will need to be extended to manage images for the new MX types. When adding an image, RingMaster will need to detect the type and manage it accordingly. When the user configures an image for a chassis, only images that match the chassis type will be shown.



### 2.3.3 IMAGE CONFIGURATION

When the user selects an image for an MX, only the images that map to the MX model should be shown. This implies that the image repository will need to support functions to retrieve images by MX model.

## 2.4 PERFORMANCE MANAGEMENT

The following sections outline the changes that will be required for MX-6 support in the area of performance management.

### 2.4.1 PERFORMANCE DATA AGGREGATION

There is no user-level change for the PM aggregation. However, there are places where the implementation will need to be updated to not assume 20+2 ports.

## 2.5 FAULT MANAGEMENT

The following sections outline the changes that will be required for MX-6 support in the area of fault management.

### 2.5.1 OPERATIONAL STATUS MONITORING

Similar to the performance aggregation, there are no user visible changes but there will be code updates to not assume a 20+2 port configuration.

## 2.6 POLICY MANAGEMENT

*NOTE: There are significant changes proposed for policy management in a different section of this document. Here for the purpose of decoupling the two features we describe how the current policy management scheme can work for the MX-6 with no changes.*

The current policy management scheme assumes that the policy is an MX-20 device. For other functions like copy-paste, import, etc. the XML mappers will need to do be flexible enough to handle applying MX-20 data to an MX-6. Hence, when a policy is applied to an MX-6 only data that is valid for the MX-6 will show up as changed. This is also how the current policy scheme works for different versions.

Note that even with all of the proposed policy changes, we will need to handle cases where a user defines a VLAN policy without policy-criteria (in which case it would need to be based on a superset of all possible configurations.)

## 2.7 RF PLANNING

As part of the design constraints, RingMaster RF Planning will require the user to now select the appropriate chassis type they wish to deploy in their network.

The algorithms which try to determine and allocate ports now must take account of the chassis type and the various port configurations.

*At this time we are assuming that there is no requirement for RingMaster to try and recommend a chassis type. Need to verify this.*

## 2.8 RF DETECTION

RF Sweeps may be considerably different when the MX-6 ships due to the lower-cost hardware and limitations. Therefore it is expected that there may be additional software required in RF detection control and configuration for this new hardware. At a minimum, the RF detection wizards and results pages must be able to handle different chassis types and possibly results.

## 2.9 REPORTS

The following sections outline the changes that will be required for MX-6 support in the area of reports.

### 2.9.1 NETWORK TOPOLOGY VERIFICATION

Network topology verification is an important feature in RingMaster that becomes more important as different chassis types can exist in the network. Verifying that the MX type matches the network plan...etc is an important extension of this logic.

### 2.9.2 INVENTORY REPORT

The inventory report will need to identify the MX type for all chassis listed.

### 2.9.3 WORK ORDER

RingMaster work-order generation will require additional features to show the user chassis types as part of the report.

## 2.10 ESTIMATES

Task	Estimate (Days)
DP Simulator changes (boot status, etc.)	1
Model Changes, Versioning and MX Creation	5
MX Upload	2
Deploy (+button/UI changes)	2
Distribute Image/Config (+button/UI changes)	2
Devif updates to retrieve & cache MX type.	1
Change Management (Accept changes)	1
XML Mapping to map ports across devices	2
Copy & Paste/Paste-Replace	5
Import & Export	4
Verification on Export	3
Image Management	3
PM & Oper Status	2
RF Planning – port allocation	1
RF Detection	??
Network Topology Report	2
Inventory Report & Work Order	2

---

### 3 DISTRIBUTED MP (DMP) SUPPORT

---

#### 3.1 OVERVIEW

The significant change to the management model for support of MX/MP separated by an intervening L2 or L3 network is related to the pre-configuration steps the customer now has to perform.

This section will define the changes to and impact on RingMaster with the introduction of Distributed MP (DMP).

##### 3.1.1 FEATURE SUMMARY

The complete details of feature requirements are in the

[http://intranet.trpz.com/highwire/productmgmt/PDD/v2.0/IntermediateL2\\_L3.pdf](http://intranet.trpz.com/highwire/productmgmt/PDD/v2.0/IntermediateL2_L3.pdf)

The goal is to integrate Distributed MPs into RingMaster seamlessly ensuring all features of RingMaster function normally.

To summarize the features in RingMaster perspective:

- Ability to configure a DMP
- Ability to configure n- Redundancy for a MP
- Ability to plan with incomplete DMP configuration
- Ability to update the MP configuration from its announce status
- Ability to monitor MP that has at least one “indirect-connection”

##### 3.1.2 DISTRIBUTED MOBILITY POINT (DMP)

A DMP is an access point connected to a MX port with an L2/L3 network in-between them. It is the ability to allow users to place Access Points in remote locations where the Ethernet cable length limit of 100 meters is an issue.

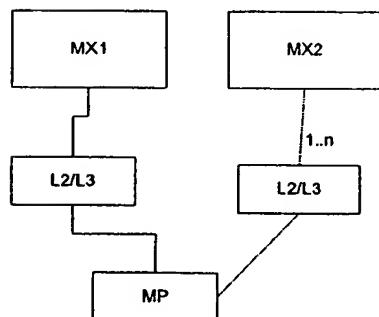
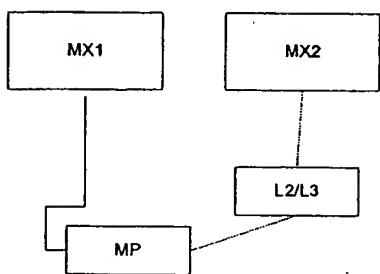
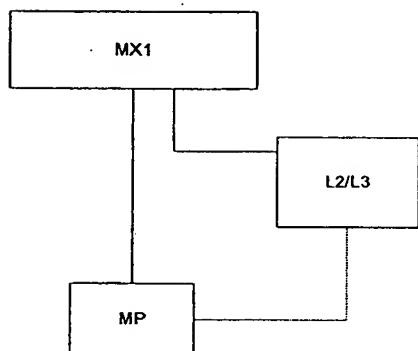
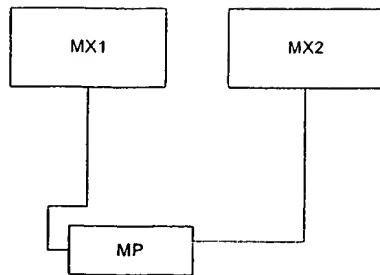
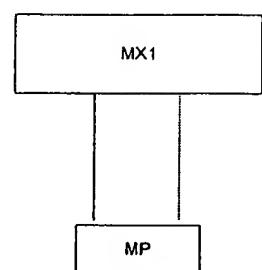
Simply put, an Access Point is a DMP, if it is not directly attached to a physical port;

Typical usage of DMP is to place in remote offices, where 1 or two access points are required and it is dangerous or management nightmare to place a MX in a wiring closet.

DMP is just like another MP. RingMaster will not distinguish between them, except when it comes to deploy such configurations as certain information will be required for MP to receive its configuration from the MX. The user will continue to define an Access Point, the way it is currently defined. The only difference being that the user needs to make it clear when configuring MP on a MX, if it is connected to a physical numbered port or if it is a DMP.

In this document, a “direct-connect” indicates that the MP is directly connected to a MX port. And similarly, a “indirect-connect” indicates that the MP is indirectly connected to the MX via a L2/L3 network

The types of connection a MP can now have are as follows:



### 3.2 USE CASES

The two ways of configuring MPs, i.e.; RF Planning tool or manual MP configuration will continue to be supported with the introduction of DMP. As always, creating new MP configurations is recommended by using the RF Planning tool.

In order for RF Planning tool to be able to generate MP Configurations, it will need some information in addition to what is requested in the current release.

#### 3.2.1 USAGE OF RF PLANNING TOOL

The work flow of using RF Planning tool will be as follows:

1. User draws coverage area and provides its details.
2. User manages Design Constraints on the Coverage Area before attempting “compute and place”
3. Upon Compute and Place, the RF Planning tool will create MP with “direct-connection” or “indirect-connection” to MXs. The planning tool will also add the desired redundancy to the MP.

#### 3.2.2 DEPLOYING DMP CONFIGURATIONS

1. The user creates MP with one ore more “indirect-connections” either manually or using RF Planning tool.
2. The user specifies the “mandatory” serial number for each MP that has at least one “indirect-connection”
3. The user deploys the MP configuration

#### 3.2.3 MANAGING DESIGN CONSTRAINTS

As in the current release, the user has no flexibility of managing Design Constraints per Coverage Area with 2.0; RingMaster will allow the user to manage the constraints at an area level. More details about this can be found in section Enhancements to RF Planning. The ways to manage design constraints will be as follows:

1. User clicks on Manage Constraints action
2. User applies certain constraints to selective Coverage Areas.
3. User edits a Coverage Area
4. User modifies the Design Constraints of the area independently.

### 3.2.4 CREATE A NEW DISTRIBUTED MP

To distinguish from creation of a “direct-connect” MP in Ports Wizard, the user will be able to create a DMP in a particular device. However, the user will not be able to add redundancy to the MP while creating the DMP. The user will have to edit the DMP to add /remove / move redundancy.

### 3.2.5 MANAGING MP CONNECTION INFORMATION

1. The user edits the MP
2. The user adds/modifies/removes redundancy to MP connection by adding/modifying/removing “connection information”

### 3.2.6 UPLOAD DMP CONFIGURATION

1. The user creates a DMP configuration on CLI
2. The user uploads the configuration from MX into RingMaster
3. RingMaster creates a Distributed MP for any DMP configurations found.

### 3.2.7 CLI IMPORT/EXPORT

RingMaster will be able to handle export and import of Distributed MP just like any other piece of configuration.

### 3.2.8 COPY/PASTE/PASTE-REPLACE

The user will be able to copy/paste a Distributed MP on another Distributed MP, but not on an AP object shown under “Ports/APs” folder.

### 3.2.9 REBOOT DIALOG

The user will be able to select a Distributed MP for a reboot.

### 3.2.10 INVENTORY REPORT

Appropriate changes will be done to the inventory Report to handle Distributed MPs

### 3.3 INFORMATION MODEL

#### 3.3.1 DISTRIBUTED MP

An MP with “indirect-connection” will have the following additional attribute to define a Serial Number that will define it unique in the entire network. It will have all the common attributes and restrictions that can be had on a “direct-connect” MP.

##### 3.3.1.1 *DMP ID*

DMP ID is the key of DMP. DMP is identified by an ID. The range of ID allowed for a DMP depends on MX type. The user will identify DMP configurations by this ID. Creation, modification or deletion of DMP configurations will be based on the DMP Port ID.

MX Type	DMP ID Range
MX-20	1..40
MX-6	1..8
MX-400	1..100

##### 3.3.1.2 *SERIAL NUMBER*

Serial Number, a text field, has the following properties:

1. It is not mandatory to be entered to create the MP
2. It is required to be entered before deployment of configuration, if it is a “indirect-connect” in one of its port configurations

#### 3.3.2 MOBILITY POINT

##### 3.3.2.1 *IP ADDRESS*

This is an ip address on the Mobility Point and is not configurable. However, this information will be visible in property panel of the mobility point.

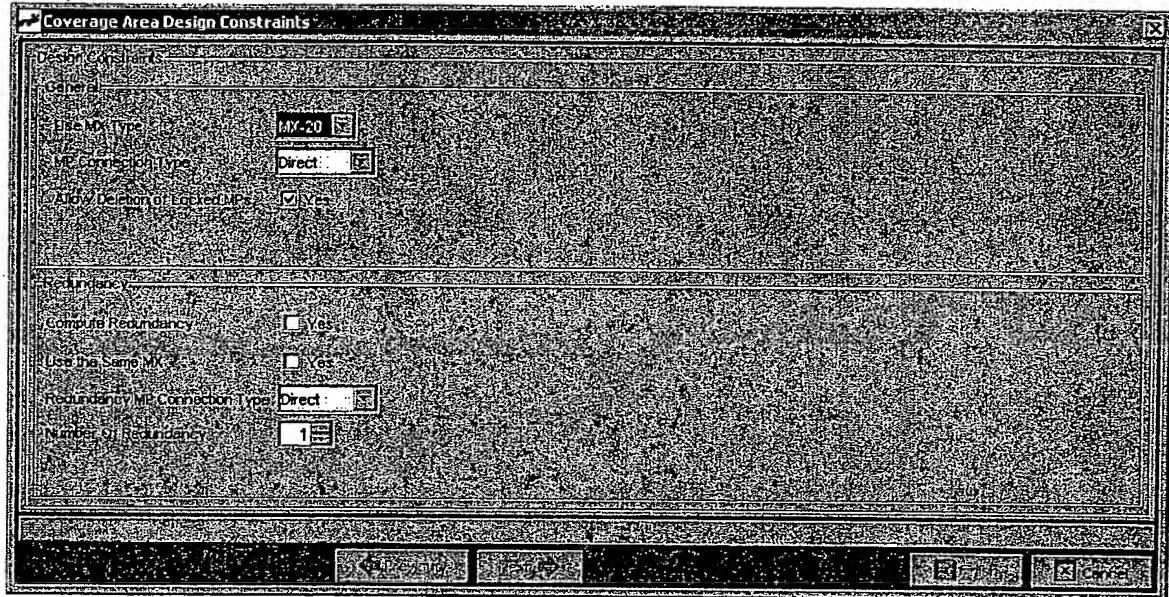
### 3.3.3 DESIGN CONSTRAINTS

The Design Constraints that can be applied on the entire floor or an individual Coverage Area. Changes to Floor Design Constraints, will be applied to any new Coverage Area that is created, unless the user applies them to a selective set of Coverage Areas.

Design Constraint	Description	Default	Comments
Use MX Type	MX-20, MX-6, MX-400 are the choices. This defines the type of MX that will be created by the planning tool	MX-20	
MP Connection Type	If any new MP is to be created, it will be created using user selection “direct-connection” or “indirect-connection” to first available port/DMPID in an MX	Choices are Direct and Distributed. Direct will be selected by default	If the MX type is MX-400, it will always be a distributed MP that will be created.
Compute Redundancy		Unchecked (no Redundancy)	
Use Same MX	If checked, the redundancy can be through the same MX from which the primary connection to MX was computed	No	
Redundancy Connection Type	When a redundancy is desired, this lets the planning tool know if the redundant connection to MP should be “direct-connect” or “indirect-connect”	Choices are Direct and Distributed. Direct will be selected by default	If the MX type is MX-400, it will always be a distributed MP that will be created.
Number of Redundant level	This is applicable only if Distributed MPs are desired for redundancy.	1	Test for 4 Max : 20
Allow Deletion of Locked MPs	Deletes the unwanted locked MPs upon compute and place	Yes	

### 3.3.4 COVERAGE AREA

The Coverage Area wizard will have an additional page to edit its design constraints. When the area is created, it gets its constraints settings from what is set on the floor.



With the introduction of distributed MP support there are no changes to current coverage area properties like wiring closet, technology type ...etc.

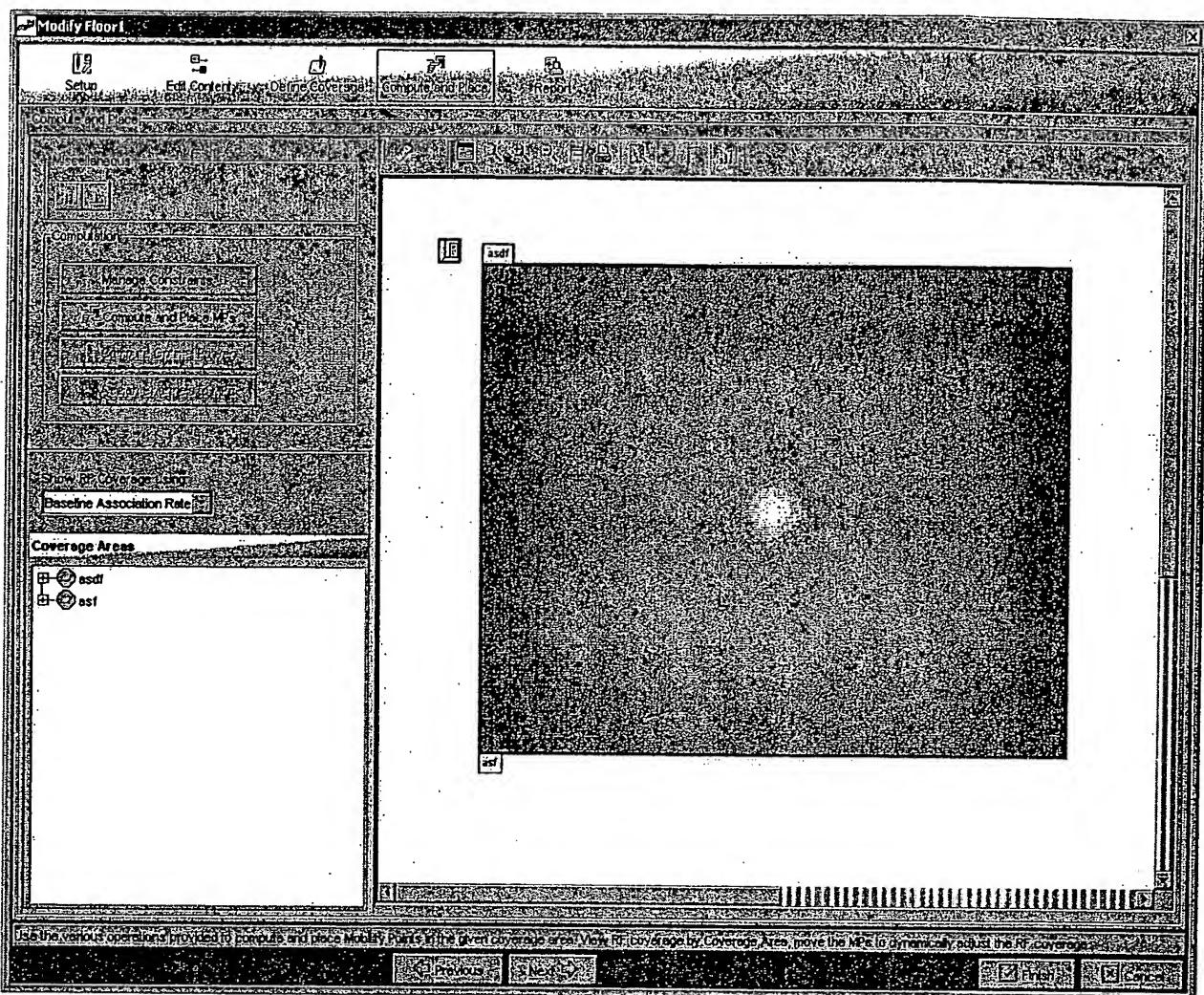
## 3.4 ENHANCEMENTS TO RF PLANNING

This section will cover the impact on RF Planning. Some constraints will be defined for the user to make it clear for the planning tool to be able to create MP.

### 3.4.1 DESIGN CONSTRAINTS MANAGEMENT

A new action will be added to "Compute and Place" page to apply design constraints at a global level to coverage areas within a floor. Current design constraints page in Compute and Place MP wizard will be removed.

# BEST AVAILABLE COPY



Clicking on Manage Constraints action will launch a wizard shown below where user can apply design constraints for all the coverage areas in a floor.

Select the constraints and the area(s) and clicking on next button will apply the constraints to selected areas and will show the progress. From the progress page user can click finish to commit the transaction or click cancel to cancel the changes. From the progress page user can click on previous button to come back and apply new constraints to different set of areas.

**Manage Constraints**

**General**

User MX-Type:

MP/Connection-Type:

Allow Deletion of Locked MPs:  Yes

**Redundancy**

Compute Redundancy:  Yes

Use the same MX:  Yes

Redundancy Connection Type:

Number Of Redundancy:

**Object Selection**

Area	Coverage Area	Selected
Area_A		<input checked="" type="checkbox"/>
Area_B		<input checked="" type="checkbox"/>
Area_C		<input type="checkbox"/>
Area_D		<input checked="" type="checkbox"/>
Area_G		<input type="checkbox"/>

<< Prev | Next >> | Finish | Cancel

**Manage Constraints**

**Status**

Applying design constraints to area\_B

Completed applying constraints to area\_B

Applying design constraints to area\_C

Completed applying constraints to area\_C

Completed applying constraints to all the selected areas

<< Prev | Next >> | Finish | Cancel

### 3.4.2 MP COMPUTATION

During computation of MP for the area, design constraints set for that coverage area is used to create distributed MP or direct-connected MP.

In case of shared areas changing design constraints of one area changes design constraints of the shared area.

**Note:** If User selects distributed MP for initial connection type / redundancy then RingMaster will select MX from the primary/redundant closet with the least DAP connections. If MXs are not available in the primary/redundant closet or if redundant closet is not provided then RingMaster will use MX in the mobility domain with least DAP connection.

### 3.4.3 WORKORDER

- New column will be added in MP table to display serial number of the distributed MP.
- Wiring closet distance table will not be generated for a distributed MP.
- For distributed MP we will display “LAN/WAN” text in MX Port column of all the tables.
- All the above changes need to be updated in both English and German version.

#### Mobility Points (MP)

MP sorted by distance from the top-left corner of the floor plan.

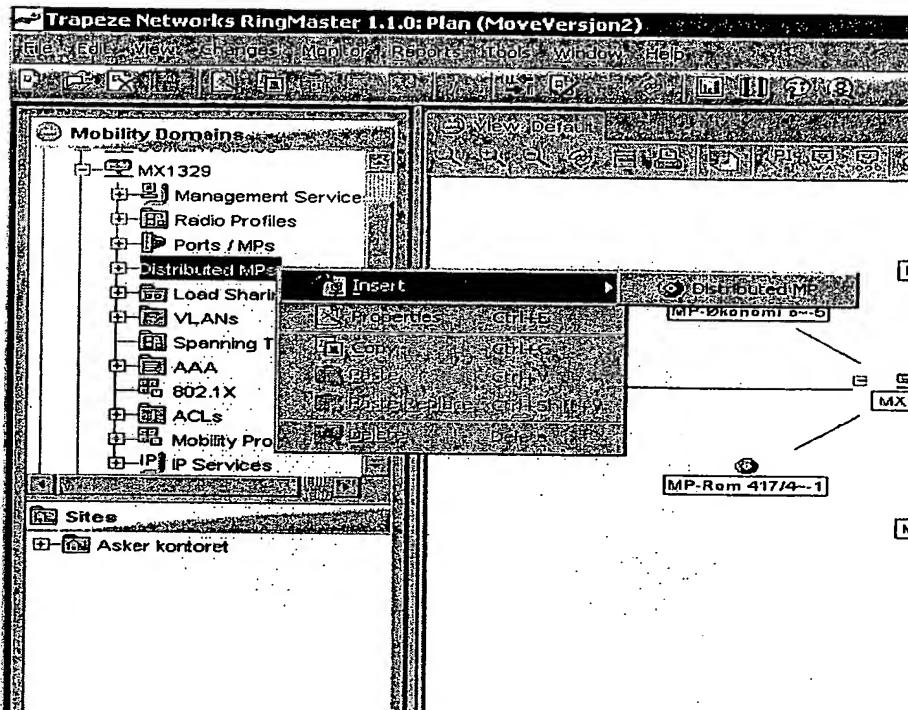
Index	MP Name	Model	MX Port (Name:Port)	MX Port (Name:Port)	Serial Number	Coverage Area (802.11a)	Coverage Area (802.11b/g)
1	MP-asf-23	MP-252	mx-104:P09	mx4394:P09		asdf	asdf
2	MP-asf-35	MP-252	LAN/WAN	LAN/WAN		asdf	
3	MP-asf-36	MP-252	LAN/WAN			asdf	
4	MP-asf-33	MP-252	LAN/WAN			asdf	
5	MP-asf-29	MP-252	mx-104:P15			asdf	

### 3.5 OTHER ENHANCEMENTS

#### 3.5.1 TREE VIEW

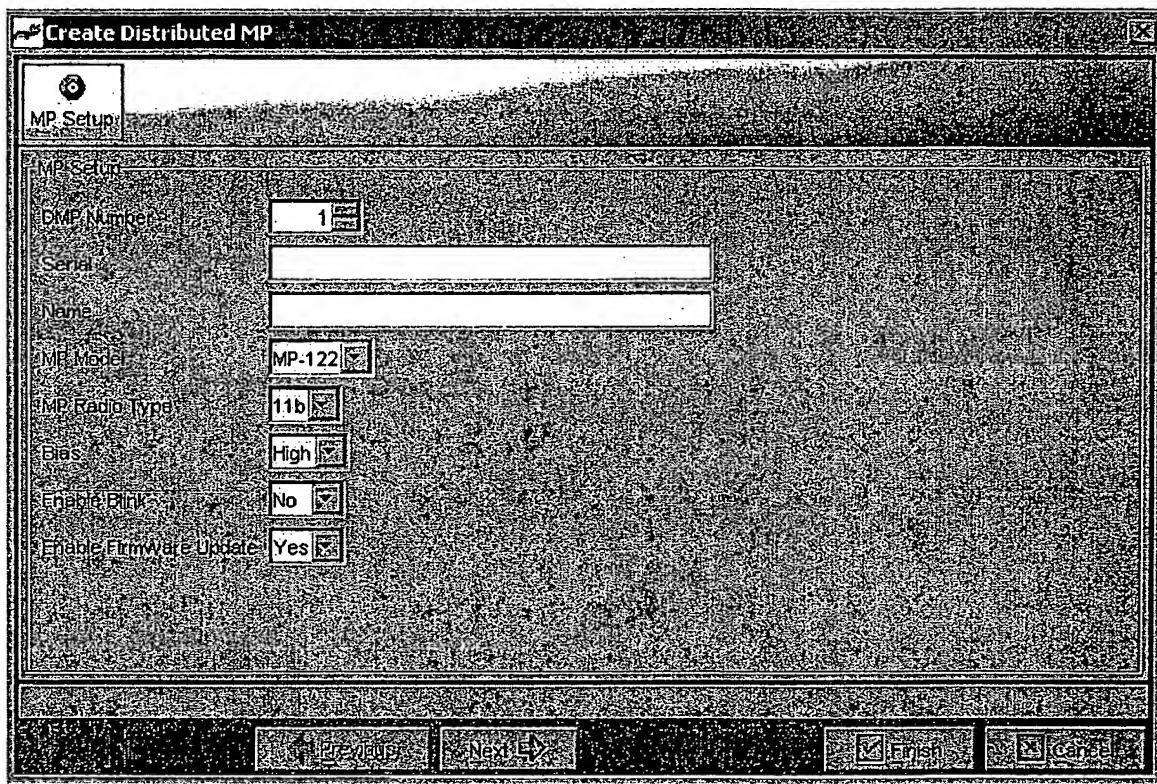
In the Devices Tree View, DMPs will appear in a separate folder under the Device. This is to visually show existence of certain DMP configurations on the MX. It will appear as follows:

# BEST AVAILABLE COPY



### 3.5.2 CREATE DMP WIZARD

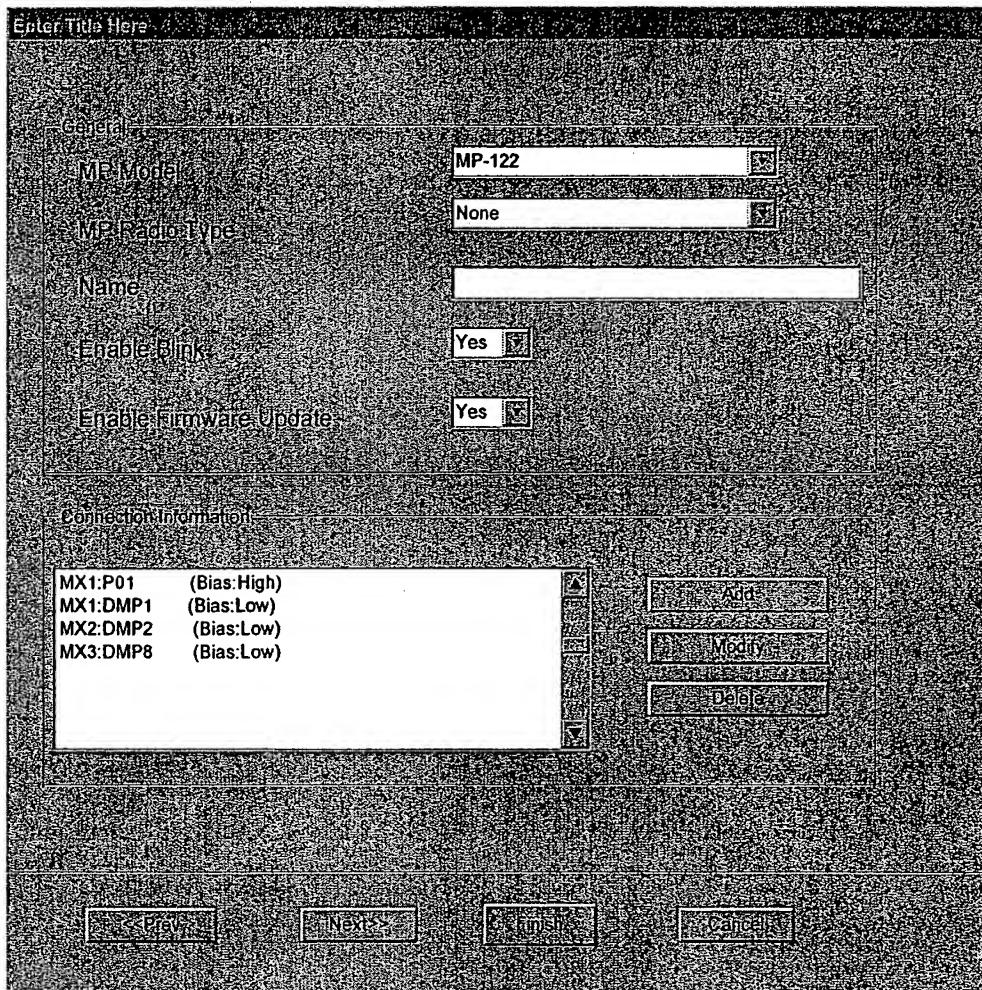
Unlike "direct-connect" MP, DMPs will have a create wizard to insert a DMP in an MX. It will allow the user to create the DMP in its basic form. To add/remove redundancy, the user will have to edit that DMP. Since RingMaster allows manual configuration of MPs, this create wizard will enable the user to create a DMP. However, it is always recommended to the user to use RF Planning tool to create the necessary MPs for their deployment.



### 3.5.3 EDIT MP WIZARD

In the current MP wizard, the GUI is restricted to a maximum of 2 port configs. With the introduction of DMP configurations, an MP can have more than 2 port configs, if the port type is DMP port.

The UI will be modified as follows to allow the user to add, modify, and delete one or more Connection information of the MP

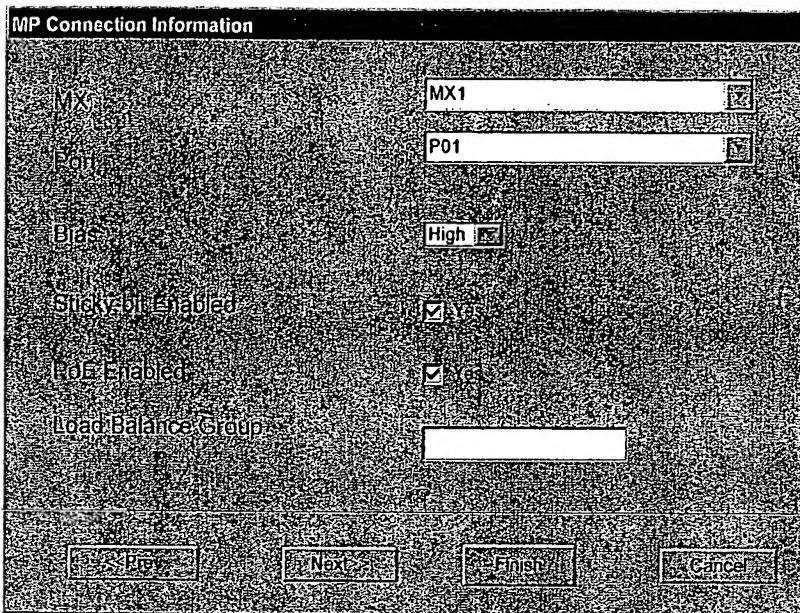


The user will be able to "ADD" two types of connections:

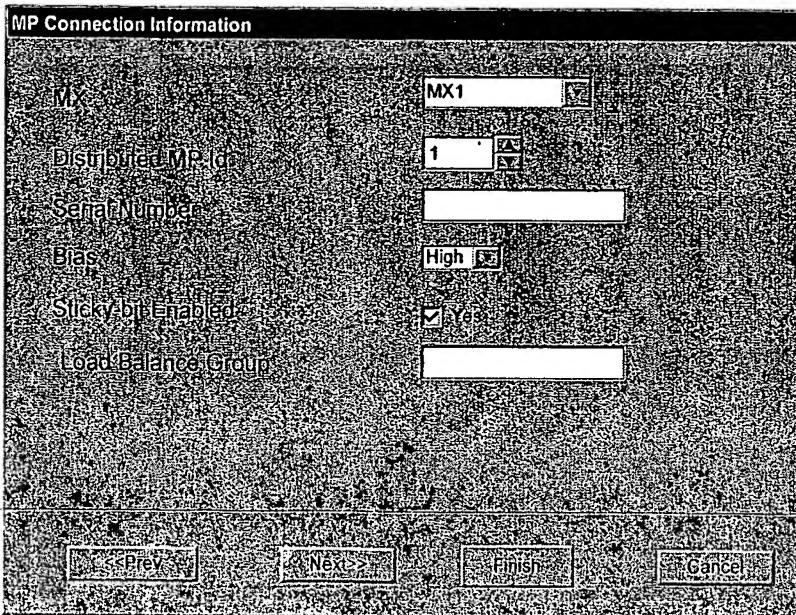
- Direct-connect (Local)
- Distributed

The user will be able to "Modify" any connection information. The user can use this feature to move the connection information within the same type of connection. For example, if the user edits a "direct-connect" connection information, the user will be able to move within any available MX ports.

When the user attempts to create/modify direct-connect connection information, following UI will be shown:



When the user attempts to create/modify distributed connection information, following UI will be shown:

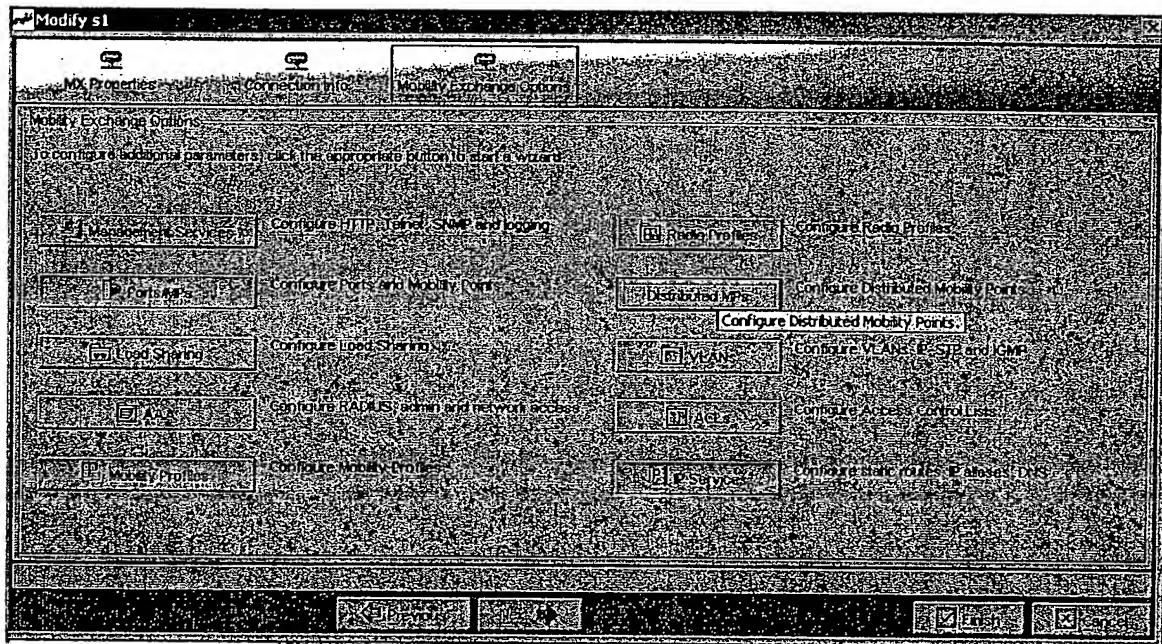


In addition to the above functionality, the WIZARD will restrict the following:

1. An MP cannot have more than two “direct-connect” connections.
2. If an MP has 2 “direct-connect” connections, it cannot have any “distributed” connections.
3. An MP can have only one “indirect-connect” connection per MX.
4. Any serial number modified here, will be applied to all “Distributed” connection information.
5. At least one connection information must be present in order to finish this wizard. The Delete action will be disabled if only one connection is remaining.

### 3.5.4 CHASSIS WIZARD

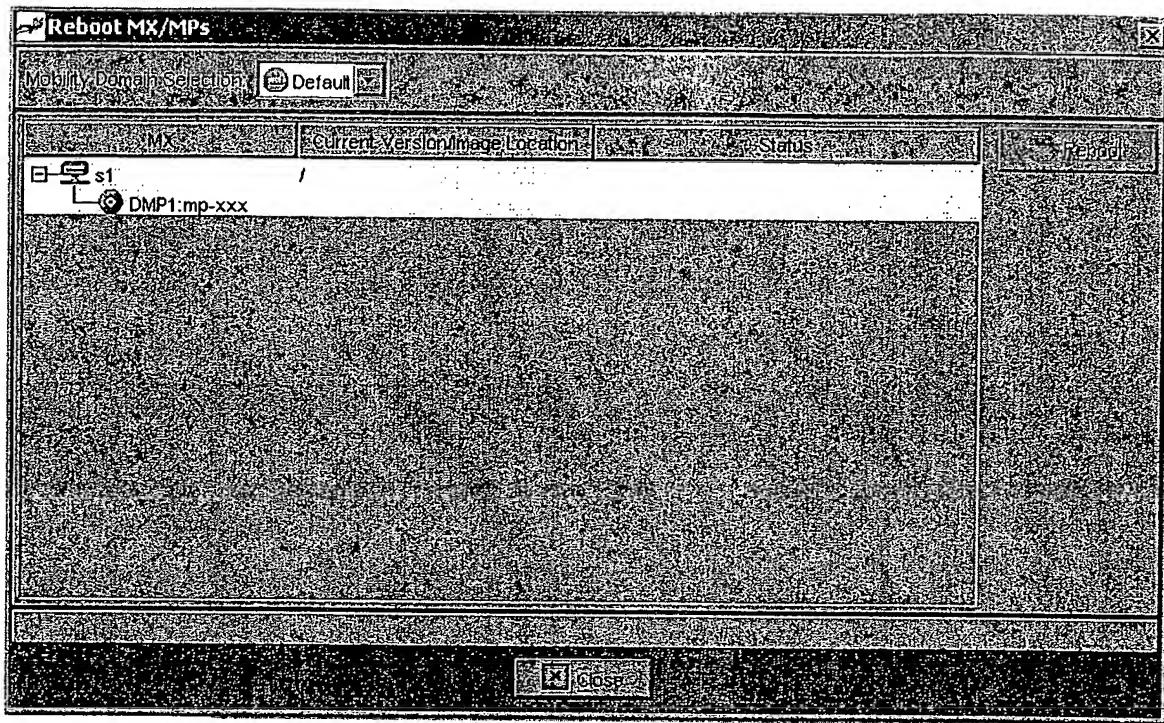
The user will be provided another action button to launch creation of Distributed MPs as shown in the following picture:



# BEST AVAILABLE COPY

## 3.5.5 REBOOT DIALOG

The user will be able to select Distributed MPs for reboot as shown in the following picture:



The user will be able to view the status of the reboot of a Distributed MP.

## 3.5.6 INVENTORY REPORT

Mobility Point Table will also include any “indirect-connections” to a Mobility Point.

The count of MP shown per MX will also consider DMP configurations within the MX

MX(s)	MP Name	Mode I	Serial Number	Bootloader Version	Radio 1 Type	Radio 2 Type	Radio 1 MAC Address	Radio 2 MAC Address
mx-104:P04								
mx-104:P06	MP04	MP-122	0321700018	QA_1.0.0.175-[REDACTED]	B	A		
mx-104:DAP1								
mx-104:P08	MP08	MP-252	0321500047	QA_1.1.0.67-[REDACTED]	G	A	00:0e:0b:00:04:d3	00:0e:0b:00:04:d4

### 3.6 FAULT MANAGEMENT

The User will be able to see the status of Distributed MPs in RingMaster. The status of MP will be a cumulative status of all its redundant configurations.

In addition to the radio status, the user will be able to see the IP address of the MP that it got assigned during its boot process. This will be visible in the Property panel of the MP. If the MP is running an image version less than 2.0, this field will have no value.

### 3.7 IMPACT ON RF DETECTION

There will be changes in RF Detection configuration page when user selects MPs to exclude all direct-connect and distributed MP will be shown.

In RF detection results page when user selects known or missing devices both direct-connect and distributed MP are considered.

### 3.8 NETWORK TOPOLOGY VERIFICATION

Network Topology Verification provides useful information to the user and can be used for the following purposes:

- Find out information about unconfigured MPs in the network
- Find out information about mis-configured MPs in the network
- Find out information about configured MPs that are not reporting any status
- Find out information about MPs that are physically multi-homed but not so in the configuration
- Find out information about an MP that has booted off an MX that had a lower Bias setting in its configuration

This information can be used by the user to update/correct the configurations in the network Plan. Currently, the user has to do it manually. RingMaster will now provide easy actions in the network topology window to correct certain kinds of information. They are:

- Ability to update Redundancy information of MP
- Ability to update Serial Number of MP

#### 3.8.1 VERIFY UNCONFIGURED MPS

An MP can be connected to the network either with “direct” or “indirect” connections, even before it is configured in the network plan and available on any one MX in the mobility domain. This rule will catch such “orphans” and notify the user.

**3.8.1.1 DIRECT-CONNECT MP**

When an MP is directly connected to MX, it is expected to have a MP configuration record at that particular port. Failure to find that record on that port where MP is requesting configuration from, will flag this MP as an "orphan". RingMaster will continue to use the existing AP-ANNOUNCE-STATUS to deduce this information.

**3.8.1.2 DISTRIBUTED MP**

When an MP is totally distributed, it is considered an "orphan" by the MX that received the configuration request, if that MX did not find any other MX in the domain to contain its configuration. It is possible that this record can move from one MX to another, if another MX is chosen by MP to request for its configuration. RingMaster will use the new DAP-ANNOUNCE-STATUS record to obtain this information

*Note: MX cluster must ensure that there is only one "orphan" record for a given serial number in any of its membership*

**Action Item: (NOS team) to verify the above note**

**3.8.1.3 MP WITH ONE DIRECT-CONNECTION AND ONE INDIRECT-CONNECTION**

For An MP that is "mixed" in connections, that is one direct and other indirect; RingMaster will use either AP-ANNOUNCE-STATUS or DAP-ANNOUNCE-STATUS to show the "orphan". Either or both records will have information about the "orphan".

**3.8.2 VERIFY CONFIGURED MPS NOT REPORTING STATUS**

This rule will catch "configured APs not reporting status" and notify the user.

**3.8.2.1 DIRECT-CONNECT MP**

When an MP is directly connected to MX, it is expected to have a MP configuration record at that particular port. Failure to find AP-ANNOUNCE-STATUS record on that port, will flag this MP as an "configured AP not reporting status".

**3.8.2.2 DISTRIBUTED MP**

When an MP is totally distributed, it is considered a "Configured MP not reporting status" when there is no DAP-ANNOUNCE-STATUS record found for that serial number.

**3.8.2.3 MP WITH ONE DIRECT-CONNECTION AND ONE INDIRECT-CONNECTION**

Failure to find any AP-ANNOUNCE-STATUS record for the direct-connection and DAP-ANNOUNCE-STATUS record for that serial number will indicate that this "configured AP is not reporting status".

### 3.8.3 VERIFY MIS-CONFIGURED MPS

This rule will catch “MP model mismatch” and notify the user. It will use the “model” information provided in AP-ANNOUNCE-STATUS or DAP-ANNOUNCE-STATUS.

### 3.8.4 VERIFY REDUNDANCY CONFIGURATIONS

This rule will catch one of the following errors:

1. MP is directly connected to two MX ports, however, in configuration, they are not redundant
2. MP is directly connected to two MX ports, however, in configuration, the MP is redundant with different port
3. MP is connected to one MX using “direct-connection”, and other MX using “indirect-connection”, and the MP is not redundant in the configuration

### 3.8.5 VERIFY SERIAL NUMBER CONFIGURATION

This rule will check for serial number configuration of an MP that has a “mixed” set of connections. Typically, if MP is totally distributed and the serial number is incorrect, it will be discovered as an “orphan”. However, if it has one direct-connection, there is a possibility that the MP boots off that MX port and it may have a different serial number in its configuration

It will be a serial number mis-configuration, if:

MP is configured to one MX using “direct-connection” and other MX using “indirect-connection” and configured with Serial Number “X”, but MP with Serial Number “Y” is connected to the above configuration. (MP is not an orphan but has a serial number mis-configuration)

## 3.9. VERIFICATION RULES

Following rules will be implemented:

1. Configuration of Indirect-connection on an MP is not supported in MX version below 2.0
2. Warn the user when coverage area is associated to a remote wiring closet when you have direct connected MPs in the coverage area
3. Generate an error if user tries to deploy a distributed MP without a serial number.
4. Generate an error if there is more than allowed MP (direct-connected + distributed) for a given MX type.
5. Generate an error if both main and redundant MX is the same for a distributed MP.
6. Warn the user if distributed MP has more than allowed redundant connections.
7. Warn the user if distributed MP is created on older version of the box that does not support distributed MP.

### 3.10 CLI MAPPING/DTD CHANGES

There will be a need to correct the CLI mappings for some commands that will have additional attributes or values.

#### 3.10.1 CLI COMMANDS

Configuration of Distributed MPs will have a separate set of CLI commands. Actual CLI Commands will be provided by Product Management.

Creation of DAP:

**Set dap <dap number> serial-number <sno> model <model> type <type>**

Modification of DAP:

All current AP commands will apply to “DAP” with the replacement of keyword “ap” by “dap”. As an example,

**Set ap 1 radio 1 channel 64 (for AP connected on port 1)**

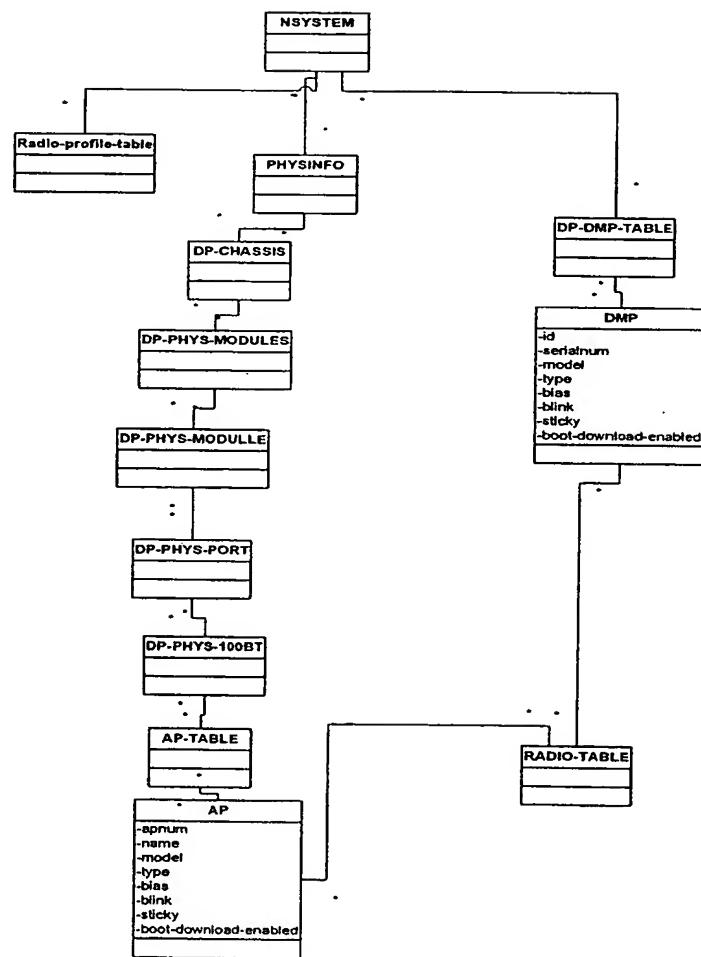
**Set dap 1 radio 1 channel 64 (for dap configured on MX on dap number 1)**

In addition to modify serial number of a dap, the CLI command

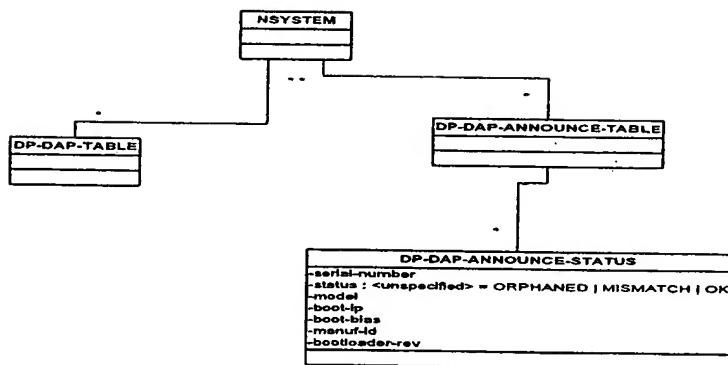
**Set dap 1 serial-number <new sno>** will be supported.

## 3.10.2 DTD MODEL CHANGES

For Configuration, following instance model will be used to distinguish between direct-connect AP and Distributed AP.



### 3.10.3 DTD CHANGES FOR ANNOUNCE STATUS



A DAP has a status of “orphaned”, if there is no config record for that serial number

A DAP has a status of “mismatch”, if the model does not match but sno matches.

A DAP has a status of “OK”, if the MP has booted off this MX and therefore, will have non-null values in boot-ip and boot-bias

### 3.11 STATISTICS

The Statistics module will be enhanced to be able to handle Distributed MPs

### 3.12 IMPACT OF MX VERSIONING

Certain CLI commands or attribute value pairs will not be available for certain versions of MX software. This will impact various planning operations.

---

## 4 POLICY MANAGEMENT

---

### 4.1 CURRENT DESIGN & ISSUES

In 1.x RingMaster the Mobility Domain Policy object is effectively an MX-20 device. This means that as new devices are created from scratch, the system would automatically clone the MX-20 policy as the basis for a new device.

#### 4.1.1 SUPPORT FOR MULTIPLE DEVICE TYPES & VERSIONS

As we introduce more device types into the supported device set in RingMaster the current scheme does not work. The policy system also does not take into account software revisions.

#### 4.1.2 POLICY CRITERIA

In 1.x RingMaster only allows a single policy per mobility domain. This is not flexible, as the user may want to pick and choose devices across mobility domains, or a subset of devices within a mobility domain, to be policy controlled.

For example, a user may want to define a AAA policy that spans all devices regardless of mobility domain membership.

Also, moving forward a user may want to define a policy that is limited to certain device types or software versions.

#### 4.1.3 CHOOSING WHAT TO APPLY

Currently the entire policy is always applied to the device to produce a diff-set that is shown as CLI commands. Then the user can select individual CLI commands to Apply to the device.

This can be dangerous also gets annoying to see unrelated changes each time the policy is applied and to have to deselect commands that are not needed. For example if the user wants to only use the policy for AAA data, they have to always deselect clearing of unrelated data like MOTD, default routes, or have to update the policy to contain that data.

### 4.2 PROPOSED CHANGES

- 1) The policy object no longer exists per mobility domain. We define a new policy database per plan. The database starts out empty, and the user can add/delete policies to the database at any point.
- 2) Policies contain set of criteria which determines whether they should be selected for a device. Initially the scope can be the device type and software version.
- 3) When a device is created or uploaded, policies with the criteria that matches the device will be selected for it. The user can fine tune this or accept all matching policies. For created devices, data from all selected policies will be applied; for uploaded devices, an association will be formed but the data will not be applied till the policy manager is invoked.
- 4) Devices and policies can be associated or disassociated at any point. The user can select a device and modify its policy associations. The user can also delete/add policies, or modify its criteria.

5) When a policy is created one or more of the following functional areas can be chosen. A functional area is any sub-tree of data in the containment hierarchy. Here is a starting list:

- a. Management services
- b. VLANs
- c. STP Properties
- d. ACLs
- e. IP Services
- f. Radio Profiles
- g. Load Sharing
- h. AAA
  - i. Radius
  - ii. Local User Database
  - iii. Admin Access Rules
  - iv. Network Access Rules
- i. Mobility Profiles

6) The user is free to select multiple policies - even with data that may conflict. This implies that there is an ordering of policies that has to be exposed to the user as if the same data fields exist in multiple policies, depending on the order of application, the results will be different.

7) The policy manager function will allow the selection of one or more policies to be applied. The resulting changes will be shown as CLI commands. The user will have the option of deselecting the entire batch of CLI commands for a particular policy, but will not be able to select individual CLI commands like is done today.

8) The Device -> Policy merge will be deprecated. Instead the user will be given a command/action to simply converting device data into policies. For example the user can select a VLAN in a device and select a menu option "Make Policy". This will launch a wizard that allows the user to create a new policy with that data, or conceivably Apply that data to an existing policy. Underneath this is the same as doing a "cut & paste" operation from a device to a policy.

#### 4.3 QUESTIONS & ISSUES

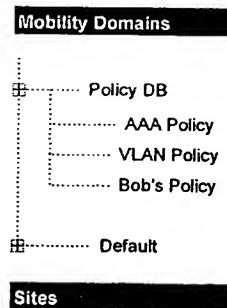
- The design and level of support for versioning will impact this and other features. Some of the questions that come up are:
  - If the user does not select any device version restriction the UI will operate in the latest version. How does this work for device types. For example the VLAN wizard for an MX-6

device will show a different number of available ports than a VLAN wizard for a MX-20. When we configure a VLAN in the policy, the type has to be known to show the available ports. Same for mobility profiles, and other objects that depend on the physical aspects of the device.

- If a policy is applied to a device that is off the wrong type, the user should be somehow shown a failure. For example, if we create a VLAN policy and add 20 VLAN members to it and Apply it to an MX-6 device, this should be flagged as an error.
- Can we somehow show a status of pending policy changes?
- The policy manager can have a drop-down list of policies and show affected devices. But this may be tedious so how can we intuitively allow the user to select multiple policies at once? We could show the changes as demarcated groups that can be selected or unselected.
- Performance improvements?

#### 4.4 CREATE POLICY

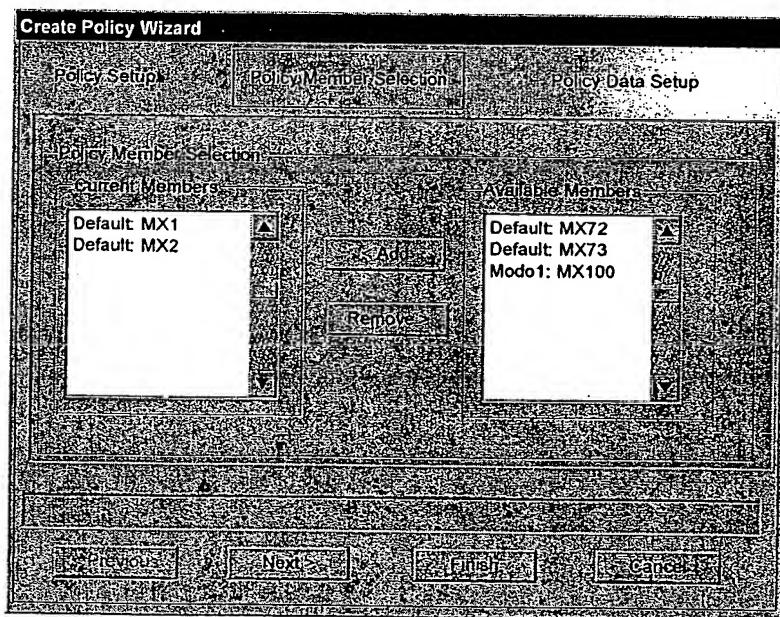
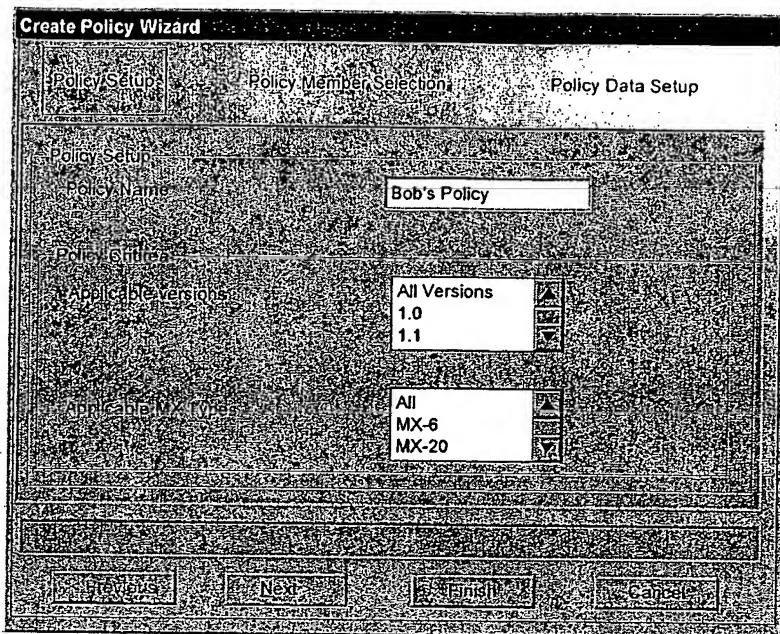
1. When a new plan is created, it contains an empty Policy DB. The user can add or delete policies to the policy database.

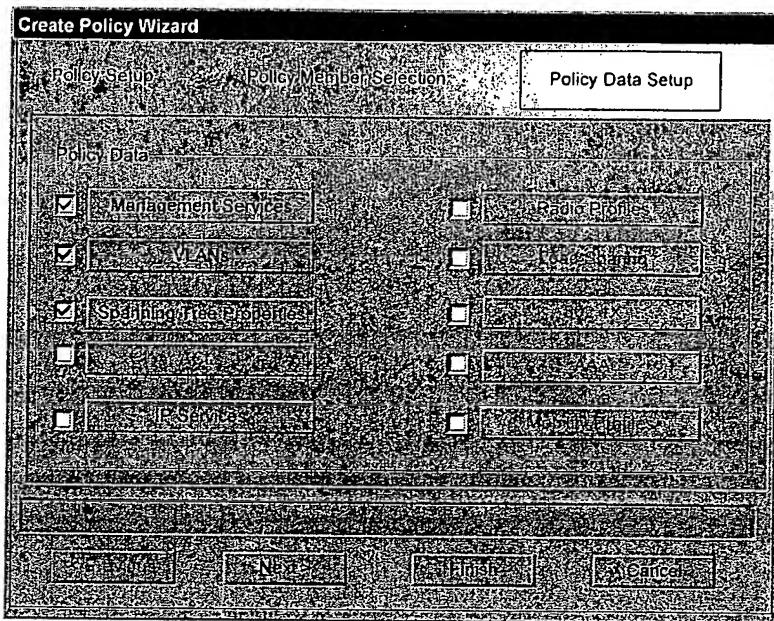


2. The Create Policy wizard consists of 3 steps:

- a. Policy Setup: here the user can name the policy and define its criteria.
- b. Device Selection: all devices that match the defined criteria are eligible to be selected. By default none are, and the user can add in the appropriate devices.
- c. Policy Data Setup: the user can select what data is to be in the policy and also launch nested wizards to configure that data.

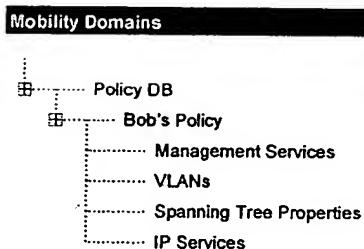
# BEST AVAILABLE COPY





#### 4.5 MODIFY POLICY

1. The user can select a policy in the organizer panel and launch a modify wizard for it. This will have the same flow as the Create Wizard. The user can also select a previously enabled configuration area under the policy and directly launch a modify wizard for that area.



2. The user can edit the policy name and/or criteria. However, if the policy has associated devices that will not match the updated criteria the user will be prompted to de-associate those devices first.

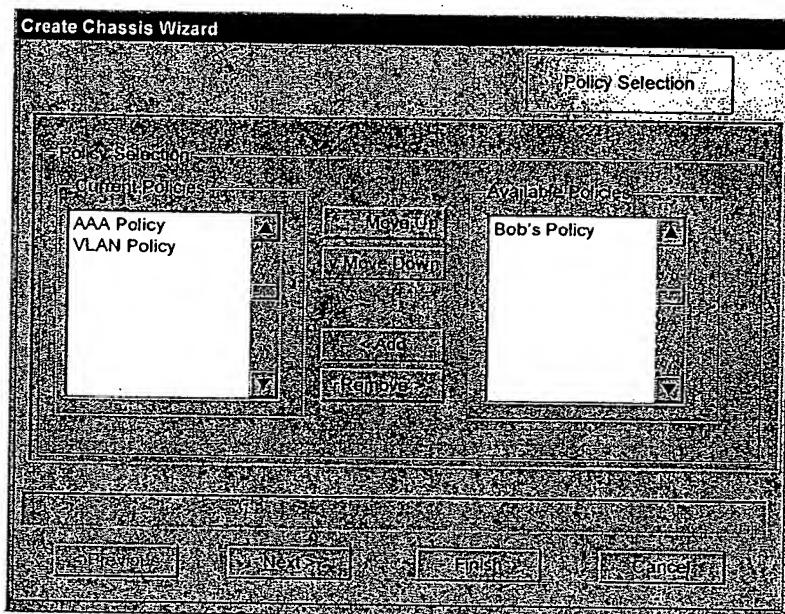
#### 4.6 DELETE POLICIES

Policies can be deleted, like any other model object.

#### 4.7 ASSOCIATE DEVICES & POLICIES

When creating a Policy it can be associated to devices. Also while creating, uploading, and/or modifying a device the user can fine-tune the policy assignments for that device.

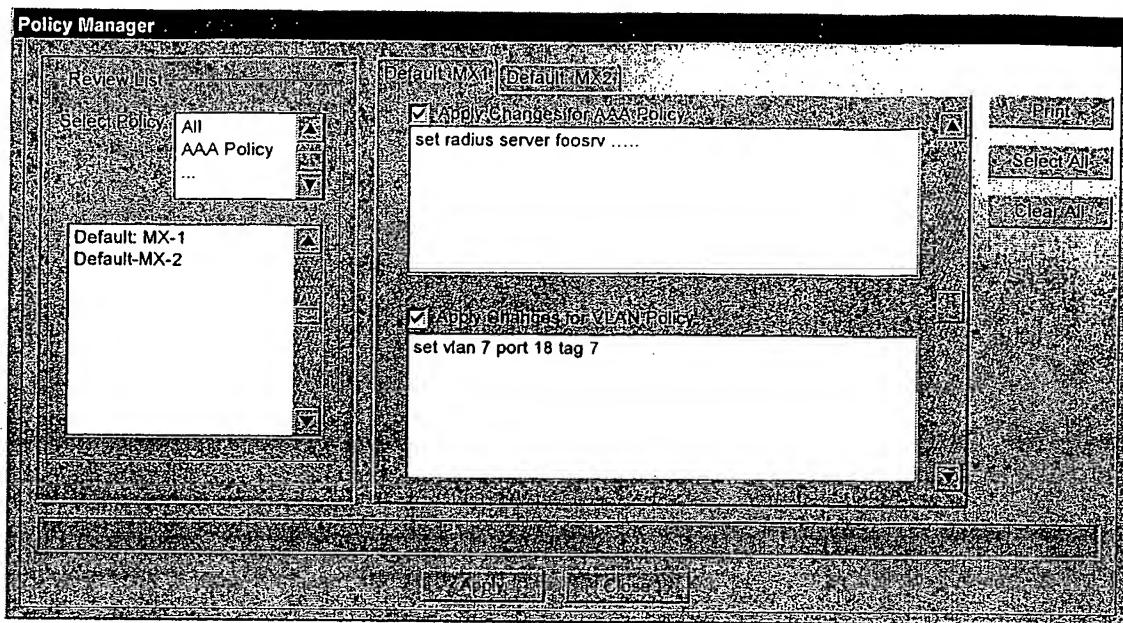
A "Policy Selection" page will be available in the Create/Modify/Upload device wizards. In this page the user can add or delete policy associations for the device. The user can also order the policies for the device. The ordering is only important if multiple policies have overlapping data.



#### 4.8 APPLY POLICIES

Policies are applied to devices using the Policy Manager.

1. User selects a single policy or all policies to be applied.
2. The list of devices with pending changes for the selected policies is calculated and displayed.
3. The user can click on an individual device and see what the result of Applying the policy is as CLI commands.
4. The user can un-select a particular set of changes (grouped by the applied policy.)
5. The user then clicks Apply. The list of changed devices is re-calculated and displayed.



#### 4.9 CREATE POLICIES FROM DEVICE DATA

A user may want to make a device's configuration data into a policy. This can be done by creating a policy and the doing a copy-n-paste of the data. It would be nice to provide a one step operation to do that.

1. User selects a device configuration element. If this is a configuration element that is policy enabled (i.e. not a port/MP, etc.) and is a policy sub-tree the user can select a menu option to make a policy out of the data.
2. A wizard will prompt the user to as if they want a new policy, or to add the data to an existing policy. The wizard will guide the user through the remaining steps to setup the policy (similar to create/modify policy.)

#### 4.10 READING OLD PLANS – BACKWARD COMPATIBILITY

The release which implements this feature will need to support the reading &conversion of network plans that have the old policy hierarchy.

The conversion will be done as follows:

1. For each mobility domain in the old plan, a policy will be created in the new plan, under the policy database, with the name: "<Modo Name> Policy".
2. The new policy will have associations with all members of the mobility domain it was created from.

This conversion will most likely be implemented in an XSLT stylesheet and will be plugged in to the overall version conversion framework (see section on Versioning.)

#### 4.11 IMPROVE PERFORMANCE

There have been complaints on the performance of (or lack thereof) the Policy dialog, and the underlying CLI mappings. Here are some things that can be done:

1. Profile to/from CLI code to identify any bottlenecks.
2. Scrub all CLI mappings to optimize how XPATH is used.
3. Allow to-CLI output to be streamed rather than wait for all commands to be generated.
4. See if XML->CS->CLI->XML->MODEL algorithm for the policy manager can be simplified. We can still use the CLI as a display but can internally Apply the XML which would avoid a CLI->XML conversion.

#### 4.12. DELIVERABLES & ESTIMATES

Here are some rough estimates (development & test) for the deliverables:

1. Model changes - Create/modify/delete policies: 4 days
2. Associate devices & policies: 2 days
3. Apply policies: 4 days
4. Multiple version & device type support: 4 days
5. Reading old plans - backward compatibility: 3 days (2 days for f/w + 1 day for policy)
6. Create policies from device data: 2 days
7. Performance Improvement: 2 days

#### 4.13. NOTES ON IMPLEMENTATION

1) There are various ways we could implement the policy rules. One way would be to actually create a new policy class that contains the criteria part:

- a list (would be individual Booleans actually) of device types (out of the descriptor map)
- a list (again would be individual Booleans) of software revisions.

The action/data part would be stored as an XML fragment of configuration. This means that the XML fragment(s) would not be completely parented (i.e. belonging to a device). For the AAA example, the root object would be the AAA class and would be owned by the policy object itself. Maybe a one-one tightly coupled relation would work so that if we were to delete the policy rule it would delete the configuration fragment. There would also be various other modifications to the system to allow such objects to be edited reusing the same pages/wizards without requiring a complete device hierarchy to support the configuration element.

2) In 1.0 each object or device that is controlled by a policy has a pointer to the policy object. We would have to make sure that after pushing a policy to a device that the particular object itself is tied to the policy rather than assuming the whole device is controlled by a single domain policy.